

# System and method for facilitating communication using presence and communication services

## Field of Invention

5

The present invention relates to communication systems and more particularly to a system for facilitating communication using separate presence and communication services.

## 10 Background of the Invention

Presence services are primarily used for providing "availability" information of one user to another user of a system. This "availability" information includes an indication of desire and availability or willingness of the user to engage in immediate  
15 communication. In order to achieve immediate communication a Presence Service is used to convey contact information to a user's presence client and the presence client is required to use that information to determine how to connect the users. This is particularly difficult when the communication services are developed by a third party vendor. Communication services include any application that creates a communication path  
20 between two or more parties. Examples of such services include telephone voice connections, Instant Messaging, chat, video conferencing, and sharing of presentations.

Existing Presence, or Instant Messaging (IM), services rely on a fixed and a built-in set of communication connection aids. For example, in order to enable communication  
25 via instant messaging services such as MSN and AOL, the communication service is normally tightly coupled with the presence service. These communication services are developed by the same vendor and usually pass through the IM server.

The prior art suffers inherent disadvantages because they are "closed" systems.  
30 There is no flexible way to add other telephony or communication services to these systems. For example, the existing Instant Messaging systems cannot add a non-IP phone call set via third party voice products. The communication connections of these systems are hardwired and limited. While this may be suitable for IM chat style of communication

services, it is not suitable for other communication services, for example, existing telephony services.

Several standard bodies are attempting to define how contact information is conveyed with a user's availability in a presence framework. Of particular interest is the IETF Instant Message and Presence Protocol (IMPP) set forth in "A Model for Presence and Instant Messaging", M. Day, J. Rosenberg, and H. Sugano, IETF RFC 2778, February 2000. Two further IETF documents discuss the Common Profile for Instant Messaging (CPIM). These two documents are the Common Presence and Instant Messaging: Message Format document (Common Presence and Instant Messaging: Message Format <draft-ietf-impp-cpim-msgfmt-03.txt>) and the CPIM Presence Information Data Format (CPIM Presence Information Data Format <draft-ietf-impp-cpim-pidf-00.txt>).

The Common Presence and Instant Messaging: Message Format document defines a mime content-type 'message/cpim' object which is a multipart entity. This document defines the structure of an IM message specifying mandatory fields such as "From:", "To:", "Date:", "Subject:", etc. The second document, CPIM Presence Information Data, defines the actual contact information that a presence client receives in order to enable a communication session between two users of a presence service. This document does not discuss how a presence client uses contact information for launching third party communication applications.

There is no flexible way to enhance any existing Presence (or Availability) Service to handle communication connection via third party communication products.

### Summary of the Invention

In accordance with the present invention, a system is provided for facilitating communication through separation between the presence and communication components. The system is based on the exchange of availability messages that are not based on the mime format, and which contain information for proper presence service functioning. Preferably, the system adopts the SLP service definition format and provides a method for a presence client to use contact information for launching third party communication applications.

Therefore, in accordance with the present invention, there is provided an approach and protocol to allow a Presence service to employ third party communication services to handle communication connection after a user's availability is established.

5

Advantageously, the present invention enables a Presence Service to deploy any number of communication mediums to create a communication path, including, for example, different Instant Messaging providers to send instant messages.

#### 10 Brief Description of the Drawings

The present invention will be described in detail with reference to the accompanying drawings, in which like numerals denote like parts, and in which:

15 Figure 1 is a block diagram of a communication system according an embodiment of the present invention;

Figure 2 is a diagram showing separation of Projection of Availability (POA) information from Control of Communications (COC) information according to the present  
20 invention;

Figure 3 is a representation of an application of the invention for facilitating ad hoc collaboration; and

25 Figure 4 is a representation of another application of the invention to contacting experts in a retail environment.

#### Detailed Description

30 A presence service generally offers two types of information, including projection of availability and communication contact information. Projection of availability includes the user's availability information and is an indication of the person's desire or willingness for communication. The availability information is projected, or conveyed, to other users.

A user appears available if that user is willing to communicate. Otherwise, the user appears unavailable.

Communication contact information includes information used for connecting with a user, such as how, where or by what means a user is available for communication. The contact information describes different communication service types on which the user is available at that time. Such communication types include telephony, Instant Messaging, chat, video streaming, etc.

The following is an example of how a presence client registers and receives availability information. This example is provided in order to give a better understanding of the mechanism by which the presence service projects a user's availability and is provided solely for the purpose of illustration. Referring to Figure 1, a presence system according to an embodiment of the present invention is indicated generally by the numeral 20. The presence system includes a presence server 22 which runs the presence service which includes a directory service 24 and a presence agent 26, a presence client 28 and resources 30, as will be described in more detail below.

The presence server 22 is a server that allows users to publish their availability to communicate to other users. The presence service on the presence server 22 includes a directory service 24 and a presence agent 26. A user registration procedure is targeted primarily to the directory service 24, while a user log-in procedure is performed on the presence agent 26. Procedures such as user registration and user log-in are two distinct operations.

25

The directory service 24 is static data storage for storing a data record in association with each user. The data record includes static information about the user including, for example, name, ID, password, address, etc. The directory service 24 allows authorized access to the system from applications other than the presence service on the presence server 22.

30

The presence agent 26 is an entity within the presence server 22 that is associated with one individual user. The presence agent 26 is responsible for storage and management of user defined presence policies, and a user profile. The presence policies

are used to determine the acceptance of “Subscription” requests from other presence agents that want to view the availability of the individual user and to control when availability notifications are to be sent to those presence agents already subscribed. It maintains contact and user state information that is projected to those users subscribed to the individual user’s presence. Communication between the user and presence agents 26 is enabled via presence clients 28.

The presence client 28 is a platform specific software entity that resides on an individual device and operates as a user interface to the presence server 22. There can exist several presence clients for an individual user. The presence client 28 facilitates communication between the user and the associated presence agent 26. This communication includes, for example, subscription and notification policies, resource contact information, user role, relationship, and buddy list management, as discussed in greater detail with reference to applicant’s copending application filed on the same day as this application and entitled “Architecture and Implementation for Control of Context Aware Call Processing with Local Feature Definition”. The login procedure is used to authenticate a client in order for the client to obtain permission for such communication.

A resource 30 is an entity that represents a communication identification and communication service. A resource can be activated/deactivated, added, removed or modified. The system 20 generally includes more than one resource 30 and each resource 30 has an associated resource profile. The resource profile includes a resource identification, an indication of whether or nor the resource is active and a communication address. An exemplary resource profile for a desktop phone is listed below.

```
<resource_profile
  id='desktop phone'
  active='inactive'
  service_url= 'service:telephony:tel/E164/4567;phone-context=+16135922122'/>
```

When a new resource 30 is added, the presence client 28 discovers the resource profile of the resource 30 and sends a request to add this resource to the presence agent 26. Once added, the resource 30 can be activated by sending the appropriate request to the presence agent 26. The resource activation procedure triggers the user’s current presence policies and publishes the user’s availability to other presence agents 26.

A user that requests to view another user's availability is referred to herein as a watcher while the user being watched is referred to as a presentity. These definitions conform to the definitions used by the IETF IMPP model discussed above. A watcher issues a subscription request to the presence server 22 in order to determine the availability of a presentity. The subscription request includes both the presentity's and the watcher's identity as well as a presentity role discussed in greater detail in applicant's co-pending application filed on the same day as this application and entitled "Role-Based Presence Enabled Service for Communication System". An exemplary XML description of the subscription request or "subscribe" message is included below. It will be understood that messages in the system 20 are referred to as events of particular types. Thus, the subscribe message is an event of type "subscribe".

```

<event
  type=subscribe
  originator=watcher@domain.com
  receiver=presentity@domain.com
  role=role />

```

After issuing a subscribe message, the watcher receives one of the following two replies.

Positive Reply	Negative Reply
<pre> &lt;reply   event=subscribe   from=presentity@domain.com   to=watcher@domain.com   value=success   role=role   key=key/&gt; </pre>	<pre> &lt;reply   event=subscribe   from=presentity@domain.com   to=watcher@domain.com   value=failed   role=role   reason=string/ </pre>

These replies include "role" and "key" tags. The role tag is used to define the role that the presentity must be in, in order for the watcher to receive a notification of the presentity's availability. The key tag is used as a mechanism for identifying the appropriate presentity's availability for a particular subscription request.

A "notify" event is used for notification of a presentity's availability. The notify event is summarized below.

```

<event
  type=notify          /* Event type */
  originator=key        /* Originator identified using key value */

```

```

value=yes | no | cancel          /* Presentity's availability */
status=status>                  /* Presentity's status */
  <resource_profile id='id' service_url='service_url'/* List of media service URLs*/
  <resource_profile id='id' service_url='service_url'/*
5  <resource_profile id='id' service_url='service_url'/*
  </event>

```

The notify event is posted if the user's presence policies for notification are triggered positively. A user's notification presence policies are triggered by changes in the user's state, location, and activation a resource. It will be understood that three availability status values are available, including "yes" for available, "no" for unavailable, or "cancel" for the cancellation of the subscription request. The contact information for the presentity is represented using a service\_url. For the present embodiment, the service\_url is the URL service scheme as defined by the IETF Service Locator Protocol (SLP), as discussed in "Service Location Protocol for Enterprise Networks", J. Kempf and P. St. Pierre, 1999, and further disclosed in the IETF document CPIM Presence Information Data Format (CPIM Presence Information Data Format <draft-ietf-impp-cpim-pidf-00.txt>).

Those of skill in the art will understand the SLP service scheme and thus, only an overview is presented herein. The SLP service scheme defines a service URL as follows: service\_url = "service:" service-type ":" service-access-info"

The service-type allows for the specification of service abstract types and particular recognized URL scheme names like http, ftp, telnet, etc. Some examples of this include "service:lpr:", "service:http:", "service:ftp:", "service:chat:". Service types can utilize a generic service name as well as the specific service name to help the user understand the URL. For example, the generic service type "printer" can be used to represent a series of protocols that relate to communication with a printer. The service in this case can be specified using "service:printer:lpr".

The service-access-info describes how the associated service is to be accessed. This includes information regarding how a user is to be contacted. Service access information consists of the following format:  
 service-access-info = "/" address-family "/" address-spec [ "/" [url-path] ["," attribute-list]]

The address-family indicates the network layer. For example “//” indicates an IP address. The address specification is the address used for communication with the presentity. A url-path is protocol-specific and specifies details regarding the manner of accessing the service. Similar to the attributes, this is an optional item. Attributes are specified in the following manner: attribute-id “=” value.

The following is a set of examples of typical IP-based service types that a presentity projects to watchers.

service:IM:ICQ//[host:port]/uin@icq.aol.com;version=1.2

10 service:chat:IRC//betty\_roland@kanata.mitel.com;version=2.3

service:http//webset2100@mitel.com;version=1.0

A non-IP protocol can also be specified for a service such as a telephony service using the SLP format (e.g. a telephony service based on the IETF RFC2806 report on URLs for Telephone Calls). The phone context allows a presence client to determine how to make a call based on the client’s location context. The example provided below shows a particular example of a phone number 4567. This phone number is a local number and therefore is only available if the user is the 16135922122 number code:

service:telephony:tel/E164/4567;phone-context=+16135922122

20

Using SLP, these services are defined using a service template and are registered with the Internet Assigned Numbers Authority (IANA) to represent the service media type.

## 25 Launching 3<sup>rd</sup> Party Communication Applications

With reference to Figure 2, after the availability information is projected, the communications contact information can be used by the presence clients 28 for establishing the communications path. Thus, the projection of availability (POA) can be de-coupled from control of communications (COC) information. The presence service on the presence server 22 is operable to manage a user’s availability and willingness to communicate while control of communications is managed by the communication services 32.



The service contact information in the availability message provides the link between the presence server 22 and communication services 32. The presence client 28 initiates the communication in one of two ways. The communication is initiated using a known interface protocol to the communication services 32 or the presence client 28 launches the communication application using a known set of parameters if no interface protocol is provided. Thus, the use of third party communication applications that can not be integrated with a presence client 28 is enabled.

Using both of these approaches, the presence client 28 employs a “mapping” that allows the presence client 28 to launch the appropriate helper application and/or objects, as described in greater detail below. This includes performing two checks prior to displaying the contact information of the presentity to the user. These checks include checking the service lookup table in order to determine if it supports the particular communication service and determining if the presentity’s domain information in the service\_url is accessible from the user’s domain. The check to determine if presentity’s contact address is accessible is based on the address-family. For an IP domain name for example, the presence client can check the domain name server (DNS) to ensure that the domain name can be mapped to an IP address. For E164 telephony addresses, this is done by using the phone-context argument that specifies the context that the telephony address is valid for.

For example, when a client receives a service:telephone:tel contact from a presence notification event it checks to see if it supports that particular service. An example of an object that supports the telephone service is presented below.

```

25      /**
        * Simple desktop telephone service component for a Mitel ICP client.
        */

30      public class E164TelService implements Runnable {

        private String resource_profile = null;
        private String serviceURL = null;
        private String resource_id = null;
        private Thread phone_monitor = null;
35      private ServerSocket socket = null;
        private boolean alive = false;
        private boolean active = false;
        private BasicClient client = null;

40      }

```

This basic E164TelService service class also needs to implement an instantiation method and connect method. An example of these are presented below.

```

5      /**
      * DesktopClient constructor.
      */
      public TelephonyResource(String serviceURL, String resource_id, BasicClient client) {
          setServiceURL(serviceURL);
          setResourceid(resource_id);
10         this.client = client;
      }

      /** Connect: This particular client uses MiTAI connections */
      public void connect(String serviceURL) throws UnknownHostException, IOException {
15
          MiTAIConnection connection = new MiTAIConnection(getMitaiHost(),
          getMitaiSocket(), Integer.parseInt(getMyExtension()));
          connection.makeCall(Integer.parseInt(getExtension(serviceURL)));
20      }

```

The service look-up table is similar to the look-up table used for Web browsers. A relationship file specifies a mapping between the contact service URL and the application that can service that URL. The service-access-info component of the service URL is specified by the service look-up table and an optional url-path is not required. The following is an exemplary format for the service look-up table.

```

table-service-access-info comm-interface
table-service-access-info = "service:" service-type ":" "/" address-family "/" comm-
interface = protocol-API | executable-file
30 protocol-API = Any alphanumeric name
executable-file = Any alphanumeric name + ".exe"

```

For the telephony example above the entry in the lookup table would appear in the following manner.

```

35      service:telephony:tel/E164/ E164TelService

```

The example above was particular to the case where a protocol API is available. In some situations a communication client can be launched for the user. Information regarding which client application to launch is required for communication using an AOL IM client, for example. This information is specified in the lookup table by referencing an executable rather than a module. For example:

```

service:chat:IRC//kanata.mitel.com IRCChat.exe
service:IM:ICQ//uin@icq.aol.comAOL.exe

```

The first example shown demonstrated the specification of using the Mitel MiTAI APIs to service the tel protocol by the presence client 28. The final look up table examples show that the ICQ service is better managed using an executable file named "AOL.exe". Thus, in this example, the presence client launches that particular application with the address contact information as a parameter.

A more complete understanding of the invention can be obtained by reference to the examples of Figures 3 and 4, which show uses of the system for making a presence server aware of communication services. These examples are provided solely for purposes of illustration and are not intended to limit the scope of the invention.

Referring to Figure 3, several users are shown for the purpose of setting up a conference call and these users desire access to other communication services in an ad hoc fashion. Automatic registration of the users' portable computers 34 is performed by creating a personal area network 36 with a base station 38 that acts as a telephone conference unit.

In operation, the users first register on to the presence service running on the presence server that is accessible to all the members of the conference call. The procedure of registering creates presence agents and sets the user profile that includes contact service information. Typical contact URLs that are shared among the members of the conference call allow them to commence chat, presentation sharing, and/or video sessions among themselves.

Reference is now made to Figure 4, which shows an application of the invention to a retail environment having a number of experts. This example leverages the use of roles for finding experts that is discussed in greater detail in applicant's co-pending application filed on the same day as this application and entitled "Role-Based Presence Enabled Service for Communication System". An expert registers with the presence service on the presence server 22 for a particular role that reflects the expert's area of expertise. Through this action he/she thereby becomes available to potential customers in a retail store via multimedia communication devices 40, such as a Mitel® Webset™ device. Typical service

URLs that can be projected to the Webset device allow for local telephony communications and sharing of product images.

Variations and modifications of the invention are possible. For example,

- 5 All such modifications and variations are within the sphere and scope of the present invention as defined by the appended claims.